



## Haga lo que hicimos: RestControllerAdvice

¡Ahora está contigo! Realice el mismo procedimiento que hice en clase, creando una clase responsable de manejar las excepciones que pueden ocurrir en las clases Controller.

### Opinión del instructor

Deberá crear una clase similar a esta:

```
@RestControllerAdvice
public class ManejadorDeErrores {

    @ExceptionHandler(EntityNotFoundException.class)
    public ResponseEntity manejarError404() {
        return ResponseEntity.notFound().build();
    }

    @ExceptionHandler(MethodArgumentNotValidException.class)
    public ResponseEntity manejarErro400(MethodArgumentNotValidException ex) {
        var errores = ex.getFieldErrors();
        return ResponseEntity.badRequest().body(errores.stream().map(
            error -> new DatosErrorValidacion(error.getField(), error.getDefaultMessage())
        ).collect(Collectors.toList()));
    }

    private record DatosErrorValidacion(String campo, String mensaje) {
        public DatosErrorValidacion(FieldError error) {
            this(error.getField(), error.getDefaultMessage());
        }
    }
}
```

```
}  
}
```

[COPIA EL CÓDIGO](#)

Además, también debe agregar la siguiente propiedad al archivo `application.properties` para evitar que el stacktracer de la excepción sea devuelto en el cuerpo de la respuesta:

```
server.error.include-stacktrace=never
```

[COPIA EL CÓDIGO](#)